

2005 International Nuclear Atlantic Conference - INAC 2005
Santos, SP, Brazil, August 28 to September 2, 2005
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR - ABEN
ISBN: 85-99141-01-5

PROGRAMAÇÃO GENÉTICA RESTRITA COM MODELO DE ILHAS APLICADA AO PLANEJAMENTO DA POLÍTICA DE TESTES DE VIGILÂNCIA DO SISTEMA AUXILIAR DE ÁGUA DE ALIMENTAÇÃO DE UM REATOR À ÁGUA PRESSURIZADA

**Rafael Pereira Baptista¹, Roberto Schirru¹, Cláudio Marcio N. A. Pereira^{1,2} e Celso
Marcelo F. Lapa²**

¹ Programa de Engenharia Nuclear, Universidade Federal do Rio de Janeiro (PEN/COPPE/UFRJ)
Cidade Universitária, Centro de Tecnologia, Ilha do Fundão, Bloco G, sala 206
Cx. Postal 68.509, 21945-970, Rio de Janeiro, Brasil
rafael@ien.gov.br

² Instituto de Engenharia Nuclear (IEN / CNEN - RJ)
Caixa Postal 68550
21945-970 Rio de Janeiro, RJ
cmnap@ien.gov.br

ABSTRACT

Este trabalho foca a apresentação de uma implementação do Modelo de Ilhas para programação genética restrita (CGP) aplicado com sucesso na solução do problema de otimização da política de testes periódicos de vigilância do sistema auxiliar de água de alimentação de um reator à água pressurizada (PWR). Para tanto, é utilizada a modelagem genética, bem como a plataforma computacional de otimização (SisGP) desenvolvida e apresentada em trabalhos anteriores. É também acoplado a esta plataforma, o Modelo de Ilhas disponível como opção na ferramenta de programação genética restrita CGP lil-gp 2.1;1.2.

O objetivo principal deste trabalho é apresentar resultados que demonstrem a eficiência e consistência do novo modelo proposto comparando-se resultados com aqueles anteriormente obtidos pela modelagem simples (que não utilizava Modelo de Ilhas) do problema em CGP.

Neste estudo utilizou-se o problema de otimização da política de testes periódicos de vigilância do sistema auxiliar de água de alimentação de um PWR apresentado por Lapa *et al.* em uma modelagem utilizando algoritmos genéticos.

Quanto ao Modelo de Ilhas utilizou-se topologia em anel com estratégia de migração síncrona e periódica.

Os experimentos realizados apontam para uma superioridade do CGP com Modelo de Ilhas, que demonstrou maior consistência e eficiência quando comparado ao CGP com a modelagem simples.

1. INTRODUÇÃO

Programação Genética (PG) [7] é uma poderosa técnica da Computação Evolucionária (CE) inspirada pelos princípios de seleção natural e evolução de espécies aplicados a um conjunto de programas de computador com o objetivo de adequá-los a solução de uma gama significativa de problemas, incluindo problemas complexos de otimização encontrados na engenharia.

Neste trabalho, busca-se estender as pesquisas de otimização das políticas de testes periódicos de vigilância do sistema auxiliar de água de alimentação (AFWS) de um reator à

água pressurizada (PWR) [2] com a adoção de uma nova modelagem, utilizando um modelo de ilhas associado à técnica de Programação Genética Restrita (CGP) [3]. Em trabalhos anteriores [9,10], apresentou-se um modelo computacional integrado (SisGP), baseado em CGP que aplicado ao problema de otimização das políticas de vigilância do AFWS obteve resultados bastante satisfatórios, porém, um grande esforço computacional foi demandado para sua aplicação. Aqui, apontam-se alguns melhoramentos na modelagem do SisGP investigando o uso de um Modelo de Ilhas na CGP com o intuito de melhorar a qualidade e eficiência do processo de otimização.

Um novo modelo computacional integrado é desenvolvido (ISisGP) utilizando a ferramenta de programação genética CGP lil-gp [3]. Para o modelo paralelo utiliza-se topologia de ilhas em anel com estratégia de migração síncrona e periódica.

O objetivo principal deste artigo é apresentar, através de comparações, vantagens do ISisGP sobre o modelo simples (não-paralelo) do SisGP para a otimização das políticas de teste de vigilância do AFWS. Para tanto, vários experimentos foram realizados e os resultados demonstraram uma superioridade do ISisGP em relação ao SisGP, não só no tempo global de otimização mas principalmente nos resultados da otimização, também ratificando os estudos do modelo em trabalhos anteriores.

2. A PROGRAMAÇÃO GENÉTICA RESTRITA COM MODELO DE ILHAS

2.1. Programação Genética Restrita

O paradigma da Programação Genética foi desenvolvido por John Koza [7] com base nos trabalhos de John Holland em Algoritmos Genéticos [6]. Atualmente representa uma área muito promissora de pesquisa em Inteligência Artificial devido a sua simplicidade e robustez.

A Programação Genética é a evolução de um conjunto de programas com o objetivo de aprendizagem por indução [12]. A idéia é ensinar computadores a se programar, isto é, a partir de especificações de comportamento, o computador deve ser capaz de induzir um programa que as satisfaça [7]. A cada programa é associado um valor de mérito (*fitness*) representando o quanto ele é capaz de resolver o problema.

Diferente dos AG clássicos, os indivíduos de uma população em PG não são cadeias de caracteres binários de comprimento fixo, mas sim programas de computador usualmente com uma estrutura em forma de árvore.

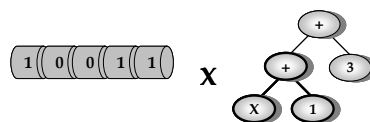


Figura 1 – População padrão do AG x PG.

Por manipular programas diretamente, a Programação Genética lida com uma estrutura relativamente complexa e variável. Tradicionalmente, esta estrutura é uma árvore de sintaxe

abstrata composta por funções em seus nós internos e por terminais em seus nós-folha. A especificação do domínio do problema é feita simplesmente pela definição dos conjuntos de funções e terminais [7].

A solução de um problema consiste em achar o melhor programa em um espaço de programas compostos por funções, variáveis e constantes apropriados ao domínio do problema.

Porém, os sistemas de programação genética tradicionais, comumente confrontam-se com um típico problema de outros paradigmas de inteligência artificial. Os mecanismos de busca de solução combinam dois elementos básicos: a representação, que determina o espaço da busca, e a ferramenta de busca, que efetivamente explora o espaço. Infelizmente muitas vezes soluções inválidas e/ou redundantes são exploradas [11].

O modelo de programação genética restrita sugerida por Janikow [3] é uma tentativa de se introduzir restrições que limitem implicitamente subespaços identificados. São restrições de natureza sintática, cuja expressão pode se revestir da tipificação de argumentos de funções, ou tipificação de valores retornados por funções ou terminais. Ou restrições de natureza semântica que se baseiam mais na interpretação de funções ou terminais.

Em termos práticos, ao imporem-se restrições sintáticas a um sistema de programação genética, define-se, para aquela população, que tipo de funções e terminais são compatíveis com os argumentos de uma dada função. Ao imporem-se restrições semânticas, define-se que tipo de funções e terminais são proibidos, mesmo que compatíveis, de serem argumentos de uma dada função.

2.3. Modelo de Ilhas

Várias abordagens para aumento de velocidade das implementações de PG tem sido propostas recentemente. Eles são conduzidos para duas direções: aumento da velocidade para minimização do esforço computacional da PG, e melhoramento no desempenho numérico do próprio algoritmo na manipulação de suas estruturas. Os principais modelos de Programação Genética Paralela (PGP) [8] são: *master-slave* (também chamado global ou mestre-escravo), *coarse-grained* (ou modelo de ilhas) [13] e *fine-grained* (também chamados celular ou grid) [1]. Exemplos dessas topologias para PGP são ilustrados na Figura 2. Nesta explanação, considerar-se-á somente o modelo de ilha.

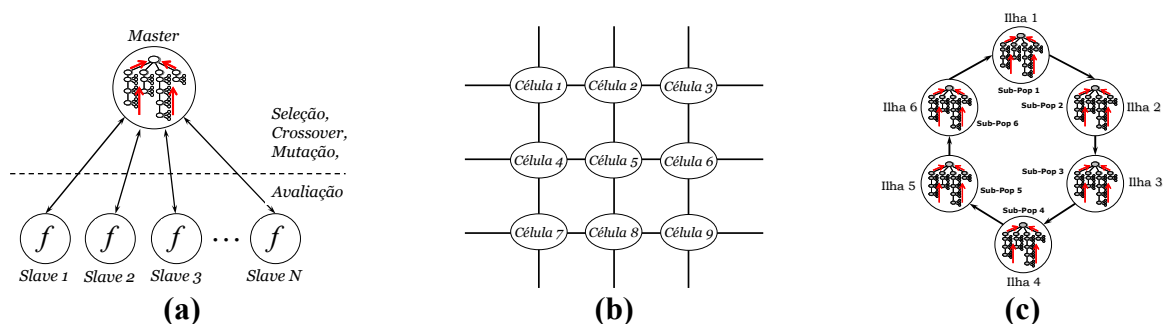


Figura 2 – Modelos de Programação Genética Paralela: *master-slave* (Fig. 2a), *fine-grained* (Fig. 2b) e *coarse-grained* (Fig. 2c).

O modelo de ilhas divide a população P de M indivíduos em N subpopulações P_1, \dots, P_N , de indivíduos M/N . Alguns algoritmos padrões de PG trabalham com cada ilha sendo responsável em inicializar, avaliar e evoluir sua própria subpopulação. As ilhas estão interconectadas de acordo com diferentes topologias de comunicação e podem trocar informações periodicamente pela migração de indivíduos de uma subpopulação para outra. O número de indivíduos a serem migrados (taxa de migração), o número de gerações depois que cada migração acontecer (frequência), a topologia de migração e o número de subpopulações são todos os parâmetros que se devem ser definidos para o método.

Em [5], uma investigação experimental da sistemática do comportamento de populações semi-isoladas em PG é apresentado. O modelo implementado consiste em ilhas evoluindo independentemente com os mesmo parâmetros de um modelo de PG canônico, exceto pela migração dos melhores p indivíduos a cada t interações de uma determinada ilha para uma outra ilha escolhida aleatoriamente, porém diferente de si mesma, onde irá substituir os piores p indivíduos. Todas as experiências mostraram que $p=10\%$ do tamanho população e $t=10$ são valores satisfatórios, e assim serão usados neste trabalho. O envio e o recebimento dos blocos de indivíduos são sincronizados. Empiricamente, foi observado em [5] que distribuir os indivíduos entre várias ilhas livremente conectadas tem a vantagem de ir além do ganho óbvio de tempo quando sistemas são executados em paralelo, verificou-se que múltiplas populações podem também conduzir a uma solução qualitativamente melhor.

3. DESCRIÇÃO DO PROBLEMA

Um dos objetivos deste trabalho é apresentar um comparativo entre a plataforma computacional integrada usando o modelo de ilhas ISisGP e a mesma plataforma com uma modelagem simples e não paralela SisGP utilizada em trabalhos anteriores[9,10] na solução deste mesmo problema. Porém, para prover uma compreensão completa deste do mesmo, será feita uma breve descrição do problema.

3.1. Descrição do Sistema

O sistema auxiliar de água de alimentação (AFWS) pode ser conceitualmente dividido em dois subsistemas. O primeiro subsistema utiliza uma única bomba turbinada (TDP) para suprir ambos os geradores de vapor (SG's). O segundo é composto por duas bombas acionadas por motores elétricos (MDP's), onde cada bomba motorizada supre água de alimentação para cada um dos SG's. Ambos os subsistemas podem ser acionados diretamente da sala de controle ou, automaticamente, atendendo a determinados sinais de segurança. Em alinhamento normal, todas as bombas succionam água proveniente do tanque de água de alimentação auxiliar (TAAA). Na perda deste, existe a possibilidade de se retirar água de fontes alternativas. Contudo, estas possibilidades de alinhamento alternativo não serão consideradas [2].

Dado que o AFWS é um sistema que durante a operação da usina encontra-se em modo de espera (*standby*), é necessário que periodicamente seus componentes sejam desalinhados para a realização de testes operacionais que poderão revelar a ocorrência de um estado de falha. Para tal, portanto, serão feitas algumas simplificações estruturais e operacionais que resultarão num esquema básico (Figura 3) contendo os componentes que estarão sujeitos aos testes operacionais para inspeção. Assim, admitiu-se que:

- i) O sistema foi analisado em alinhamento normal, não considerar-se-á, pois, a possibilidade de alimentação através de fontes alternativas (sistema de água de serviço e sistema de proteção contra incêndio) nem configurações de alinhamentos relativos aos testes periódicos ou à manutenção emergencial.
- ii) Representou-se cada grupo de válvulas, de cada uma das quatro linhas de injeção, por uma única válvula hipotética;
- iii) Admitiu-se que os componentes de cada grupo redundante fossem operacionais e estruturalmente idênticos.

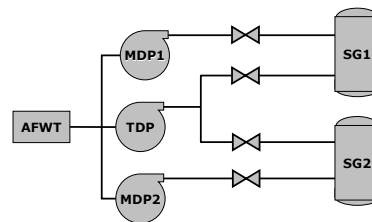


Figura 3 – Esquema Básico do AFWS

Adotando estas simplificações propostas por [2] e através da planta original do AFWS construiu-se o esquema básico simplificado (Figura 3) sobre o qual se desenvolveu o estudo descrito neste trabalho.

Em termos operacionais, a falha do sistema é o não fornecimento ou fornecimento insuficiente de água de alimentação para ambos os SG's.

3.2 Modelagem Genética

A modelagem genética segue os mesmos princípios dos trabalhos anteriores [9,10], onde o conjunto das funções é composto pela: função *RAIZ* onde é gerada uma matriz de decodificação da política de agendamentos proposta pelo candidato para o sistema e que posteriormente será passada para o Sistema Probabilístico de Análise (PSA); função *C* que representa a política de intervenções para teste de cada componente do sistema; função *t* que representa cada agendamento de intervenção para o componente; e a função terminal *T*, que tem papel semelhante à função *t*, só que também representa o final da estrutura de agendamentos para o componente. Já o conjunto dos terminais será composto por três geradores randômicos de terminais numéricos (*N1*, *N2* e *N3*) que geraram números entre 0 e 480, que é a jornada de operação em dias do sistema proposto, para estabelecer o momento exato de cada intervenção proposta na política pelas funções *t* ou *T*.

As restrições usadas para preservar a geração de estruturas válidas de agendamentos são: i) só a função *RAIZ* pode desempenhar o papel de raiz da árvore no modelo genético proposto de forma que só há uma função desta por candidato à solução; ii) a função *RAIZ* só pode ter funções *C* como seus argumentos e haverá tantas funções *C* quantos forem os componentes do sistema escolhido; iii) as funções *C* só poderão ter como seus argumentos funções *t* ou *T*; iv) as funções *t* terão 4 argumentos que devem seguir obrigatoriamente a seguinte estrutura: *galho(0)*: funções *t* ou *T*, *galho(1)*: terminal *N3*, *galho(2)*: terminal *N2* e *galho(3)*: terminal

Um exemplo de candidato à solução para um sistema fictício de quatro componentes pode ser visto na Figura 4 e o mesmo terá sua política decodificada na Tabela 1.

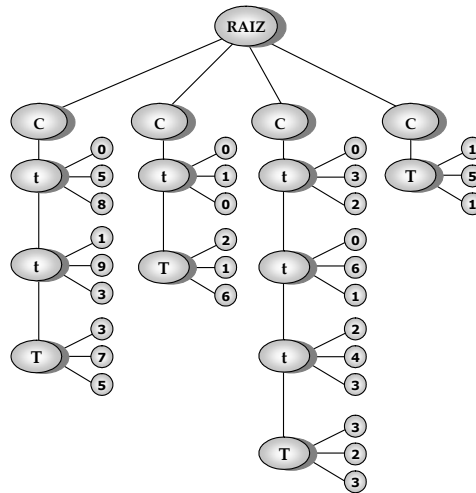


Figura 4 – Representação gráfica de um candidato a solução.

Tabela 1 – Decodificação da política de exemplo da Figura 4.

| Componente | Agendamentos dos Testes de Vigilância (dias das intervenções) |
|------------|--|
| C1 | {58, 193, 375} |
| C2 | {10, 216} |
| C3 | {32, 61, 243, 323} |
| C4 | {151} |

A Função de Aptidão (“*Fitness*”), é a média das indisponibilidades pontuais (ou instantâneas) do sistema escolhido, calculadas previamente pelo modelo probabilístico descrito por [2] considerando a política de intervenções proposta pelo modelo genético.

$$Fitness = \frac{1}{T_{miss}} \int_{t=0}^{T_{miss}} [\overline{A}(t)_{sis}] dt \quad (1)$$

onde, $\overline{A}(t)_{sis}$ é a indisponibilidade instantânea. Uma descrição mais detalhada do modelo de confiabilidade pode ser encontrada em [2].

4. EXPERIMENTOS E RESULTADOS

Todos os experimentos apresentados nesse artigo foram alcançados através da plataforma computacional integrada SisGP, que foi desenvolvida para trabalhos anteriores e que hora é agregada a uma modelagem de ilhas do CGP lil-gp. A essa plataforma deu-se o nome de

ISisGP. Tanto o código computacional desenvolvido quanto às demais ferramentas usadas nesse trabalho foram desenvolvidos em linguagem C.

O modelo genético foi implementado com base na ferramenta de programação genética CGP lil-gp 2.1;1.02 [3] e o modelo probabilístico baseado no modelo de cálculo de indisponibilidades para componente em série apresentado por Lapa [2].

Essa plataforma computacional funciona, basicamente, com a seguinte estrutura de troca de dados: A programação genética inicializa aleatoriamente o processo de criação de sua população de árvores (políticas candidatas à solução) gerando N subpopulações quantas forem o número de ilhas definidas pelo usuário para o experimento e respeitando as restrições programadas bem como os parâmetros informados. Políticas de intervenções são propostas por todas as árvores da subpopulações. Cada árvore tem sua política decodificada pela sua respectiva função *RAIZ* para uma matriz e um vetor que são transferidos para o PSA (que tem seu código replicado para paralelizar o processo de avaliação em cada uma das ilhas) que, baseado neles, calcula a indisponibilidade média do sistema (ou *Score*), retornando-o para função *RAIZ*. Esse processo é repetido pelo número de vezes igual à população inicial sugerida pelo usuário para cada ilha.

Ao final desta etapa, a programação genética, com base na *fitness* bruta (busca da menor indisponibilidade) avalia o *Score* de cada candidato à solução. Daí, utiliza essas informações para aplicar seus operadores e apontar um novo conjunto de políticas candidatas à solução até que se tenha um resultado satisfatório. Também é definido como parâmetro para o ISisGP uma estratégia de migração determinando que a cada passo de tempo definido, um determinado número de indivíduos da subpopulação serão substituídos por outros indivíduos vindos de uma subpopulação vizinha, garantindo assim maior diversidade dentro das ilhas.

Para todos os estudos de casos esses processos serão mantidos trocando-se apenas os parâmetros de execução da programação genética, rampa de profundidades da população inicial, taxas dos operadores e as estratégias de migração.

Na Tabela 2 são apresentados os valores mínimos de indisponibilidade obtidos com os cinco melhores experimentos do ISisGP, como também os obtidos pelos cinco melhores experimentos realizados com o SisGP [9,10].

Tabela 2 – Comparação entre resultados do ISisGP e do SisGP com população de 400 indivíduos

| Experimentos | Indisponibilidades Médias | |
|-------------------------------------|---------------------------|-------------------------|
| | SisGP | ISisGP ^a |
| 1 | 5.1475×10^{-5} | 5.1386×10^{-5} |
| 2 | 5.1848×10^{-5} | 5.1285×10^{-5} |
| 3 | 5.1614×10^{-5} | 5.1432×10^{-5} |
| 4 | 5.1713×10^{-5} | 5.1094×10^{-5} |
| 5 | 5.1564×10^{-5} | 5.1503×10^{-5} |
| Média Geral | 5.1643×10^{-5} | 5.1336×10^{-5} |
| Média Horária (em horas) | 50.7 | 21.2 |

^a Experimentos variando entre 2 e 4 ilhas.

Usando o ISisGP, podem-se obter níveis de indisponibilidades ligeiramente mais baixos que os obtidos com o SisGP. Essas melhorias nas taxas de indisponibilidade podem não ser muito expressiva nesta aplicação, mas se deve lembrar que se está trabalhando com sistemas nucleares, que por projeto, já tem uma disponibilidade/confiabilidade extremamente alta devido às redundâncias, exigências de fábrica, etc. Em outros sistemas industriais, onde a disponibilidade média aceitável é bem mais baixa, o impacto desses números seria bem mais expressivo, principalmente ao longo de anos de operação. Outro detalhe significativo que deve ser observado é uma redução de mais de 50% no tempo necessário para o processamento.

Tabela 3 – Política de agendamentos proposta pelo experimento 4.

| Componente | Agendamentos de teste de vigilância (dias das intervenções) |
|--------------------------|--|
| <i>V1</i> (40 paradas) | {28, 40, 52, 64, 74, 98, 118, 130, 140, 160, 174, 184, 194, 208, 220, 232, 246, 254, 272, 286, 294, 310, 320, 328, 336, 344, 354, 364, 372, 378, 388, 394, 404, 410, 420, 426, 440, 444, 458, 470} |
| <i>MDP1</i> (15 paradas) | {40, 64, 118, 160, 194, 232, 272, 294, 328, 354, 372, 394, 420, 440, 458} |
| <i>V2</i> (17 paradas) | {36, 94, 144, 172, 198, 230, 258, 284, 318, 346, 366, 386, 408, 428, 438, 456, 466} |
| <i>V3</i> (17 paradas) | {38, 92, 144, 172, 198, 230, 258, 284, 318, 348, 366, 386, 408, 428, 438, 452, 466} |
| <i>TDP</i> (10 paradas) | {76, 144, 198, 256, 302, 342, 382, 408, 428, 448} |
| <i>V4</i> (40 paradas) | {18, 34, 50, 58, 70, 88, 100, 112, 128, 142, 154, 168, 182, 192, 204, 216, 228, 240, 252, 270, 280, 290, 300, 312, 326, 338, 350, 360, 368, 380, 390, 400, 406, 416, 424, 436, 446, 454, 460, 468} |
| <i>MDP2</i> (15 paradas) | {50, 100, 128, 182, 216, 252, 290, 312, 338, 368, 390, 406, 424, 446, 460} |

5. CONCLUSÕES

Com esse trabalho, ratificou-se a importância de utilizar a computação evolucionária paralela em problemas complexos de otimização com consumo elevado de tempo, não somente pelo ganho de tempo no processamento, mas principalmente devido aos melhores resultados que podem ser achados.

Também mostrou-se que a modelagem de ilhas proposta é mais eficiente e robusta para propor políticas de testes periódicos de vigilância otimizadas para o AFWS com um tempo de processamento bastante inferior em relação aos estudos anteriores [2, 9, 10].

Os resultados do ISisGP foram ligeiramente melhores que os do SisGP, mas devem-se ressaltar algumas características do modelo de ilhas na PG que trazem grandes vantagens para essa modelagem. A PG trabalha com estruturas de comprimentos variáveis, que permitem um ajuste mais fino a soluções de problemas desta natureza e complexidade, porém acarretam em um aumento considerável do custo computacional principalmente quando aplicados a problemas maiores e mais complexos. Quando se paraleliza a manipulação dessas estruturas, pode-se ter ganhos não somente no tempo de processamento com uma redução do custo computacional necessário para o processamento de cada parte da população (subpopulações), mas também uma maior diversidade nas populações em diferentes etapas da evolução da solução do problema.

Trabalhos futuros nesta linha de pesquisa devem investigar a aplicação de ferramentas como Nichos e outras técnicas e topologias para o paralelismo.

AGRADECIMENTOS

Rafael P. Baptista é apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq.

Roberto Schirru é apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq.

Cláudio M.N.A. Pereira é apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq.

Celso M.F. Lapa é apoiado pela Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro – FAPERJ.

REFERENCIAS

1. C. C. Pettey., “Diffusion (cellular) models”, In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C6.4:1–6. Institute of Physics Publishing and Oxford University Press, Bristol, New York (1997).
2. C. M. F. LAPA, C. M. N. A. PEREIRA, P. F. FRUTUOSO E MELO, “An application of genetic algorithms to surveillance tests optimization of a PWR auxiliary feedwater system”, *International Journal of Intelligent Systems* 17 (8), 813-831 (2002).
3. C. Z. JANIKOW e S. W. DEWEESE, “CGP lil-gp 2.1;1.02 User’s Manual”, Department of Mathematics and Computer Science, University of Missouri, St. Louis. janikow@radom.umsl.edu (1997).
4. D. E. GOLDBERG, “Genetic algorithms in search, optimization and machine learning”, Alabama: Addison Wesley, 413p (1989).
5. F. Fernandez, M. Tomassini, and L. Vanneschi. “An empirical study of multipopulation genetic programming” *Genetic Programming and Evolvable Machines*, 4(1):21–51, March (2003).
6. J. H. HOLLAND, “Adaptation in natural and artificial systems” The University of Michigan Press, Ann Arbor, MI (1975).
7. J. R. KOZA, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, ISBN 0262111705, MIT Press (1992).
8. M. Tomassini. “Parallel and distributed evolutionary algorithms: A review” In P. Neittaanmki K. Miettinen, M. Mkel and J. Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, J. Wiley and Sons, Chichester (1999).
9. R. P. BAPTISTA, R. SCHIRRU, C. M. N. A. PEREIRA *et al.*, “Surveillance Tests Optimization of a PWR Auxiliary Feedwater System by Constrained Genetic Programming”, *Applied Computational Intelligence*, Proceedings of the 6th International FLINS Conference, ISBN 981-238-873-7, World Scientific, England, p. 583-588, Belgium (2004).
10. R. P. BAPTISTA, R. SCHIRRU, C. M. N. A. PEREIRA *et al.*, “Programação Genética Aplicada na Otimização da Política de Testes Periódicos de Vigilância do Sistema Auxiliar de Água de Alimentação de uma Planta Nuclear PWR”, *XXV CILAMCE Iberian Latin America Congress on Computational Methods*, Recife/PE (2004).
11. R. P. DOMINGOS, “Modelos Neuronebulosos Evolucionários Aplicados ao Controle de Processos Nucleares” **D.Sc thesis**. COPPE, Universidade Federal do Rio de Janeiro, (2003).

12. W. BANZHAF; P. Nordin; R. E. Keller e F. D. Francone, “Genetic Programming: an introduction” **ISBN 155860510X**. Morgan Kaufmann (1998).
13. W. N. Martin, J. Lienig, and J. P. Cohoon. “Island (migration) models: evolutionary algorithms based on punctuated equilibria” In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C6.3:1–16. Institute of Physics Publishing and Oxford University Press, Bristol, New York (1997).